

REMARKS

Claims 1-17 and 20 are active in the application. Claim 19 has been canceled. Claim 1 has been amended to substantially incorporate features of now canceled claim 19. Claim 1 now includes the language that the thread unit is a “hardware” thread unit. This language is of a clarifying nature, and is supported by text at page 6, lines 4-8 of the present specification, which describes the thread unit as comprising components that are necessarily hardware apparatus. Amended claim 1, due to its incorporation of features of now canceled claim 19, does not present new issues and should be proper for consideration and entry at this time. Further, amended claim 1 should now be in *prima facie* condition for allowance.

Claims 1-12 and 20 were rejected as being obvious over a combination of U.S. Patent 6,393,026 to Irwin in view of U.S. Patent 6,047,002 to Hartmann and U.S. Patent 5,353,418 to Nikhil. Claim 13 was rejected as being obvious over a combination of Irwin, Hartmann, and Nikhil further in view of U.S. Patent 6,629,257 to Hartwell. Claims 14-15 have been rejected as being obvious over a combination of Irwin, Hartmann, and Nikhil further in view of U.S. Patent 5,513,354 to Dwork. Claim 16 has been rejected as being obvious over a combination of Irwin, Hartmann, Nikhil, and Dwork in view of U.S. Patent 4,674,034 to Iwashita. Claim 17 has been rejected as being obvious over a combination of Irwin, Hartmann, Nikhil, and Dwork in view of U.S. Patent 5,428,766 to Seaman. Claim 19 has been rejected as being obvious over a combination of Irwin, Harmann, and Nikhil further in view of U.S. Publication 2001/0025234 to Gamo. Each of these rejections is overcome in view of the amendment above.

Claim 1, as amended, requires that related data frames be dispatched to the same thread unit. The requirement of using the same thread unit was specified in now canceled claim 19. Dispatching related data frames to the same thread unit tends to increase operating efficiency and speed of the protocol converter.

Gamo relates a method for data communication in which software objects perform message communication. Software objects are organized into complex objects that each contain several objects. The objects grouped in a complex object have execution

seriality and are executed on the same instruction (software) thread. This expedient tends to reduce processing time. This is because grouping the objects into complex objects reduces the need for message communication between complex objects (also known as context switching). This aspect of Gamo is described in paragraphs 0040, 0041 and 0042.

5 The Office Action incorrectly argues that Gamo at paragraph [0039] teaches that related data frames can be dispatched to the same thread unit. Specifically, Gamo at paragraph 0039 states that “objects have execution seriality so that when an object within the same complex object is invoked, message communication is executed in the same thread.” (emphasis added) This aspect of Gamo is very different from the present
10 invention, as explained in more detail below, and combining Gamo with a combination of Irwin, Hartman, and Nikhil (if such a combination could be made-as noted previously, the undersigned is of the position that the combination is not proper) would not yield or make obvious the claimed invention.

 . It should be understood that Gamo is different from the present invention in at
15 least the followingl significant ways:

1) Firstly, the objects of Gamo are not data frames. The objects of Gamo are executed serially in the same thread, but they are very different in nature from the data frames of the present invention. The objects of Gamo are necessarily executable instructions. This is clear from paragraph 39, which states that “The objects have
20 execution seriality...”. Also, it is well known that software objects necessarily consist of executable code. By contrast, data frames of the present invention are not executable code. The data frames of the present invention are merely “containers” for data, and do not have executable code. The present data frames may contain data that can be used as executable code after conversion and extraction from a data frame, but the data frames
25 themselves are merely containers for data being converted between protocols. Accordingly, execution of objects in a thread according to Gamo is quite different from dispatching data frames to the same thread unit in the present invention. The objects of Gamo are completely different from data frames of the present invention. Gamo does not teach or suggest that related data frames can be dispatched to a thread unit. Accordingly,
30 Gamo does not meet the limitation of claim 1 requiring dispatching of data frames to a particular thread unit, and the rejection of claim 1 must be withdrawn.

2) Secondly, Gamo requires message communication between the software objects. The objects of Gamo comprise executable code and the objects must be able to communicate with one another for proper execution of the code (see paragraphs 0013, 0014 for example). By comparison, data frames of the present invention are not software
5 objects and are not capable of message communication. The data frames of the present invention function merely to transfer data, and are not capable of performing communication or any other kind of activity or code execution. For this additional reason, Gamo is not applicable to dispatching of data frames, as required in the present invention, and the rejection of claim 1 as amended must be withdrawn.

10 3) Thirdly, the threads of Gamo are necessarily software or instruction threads. The threads of Gamo are not hardware threads, which are required in the present invention. For instance, Gamo describes the threads as “execution threads” in the ABSTRACT, and in paragraphs 0014, 0023, 0024, 0047, 0054, 0106, 0162, 0165 and elsewhere. Nowhere does Gamo suggest that the threads are hardware threads comprising
15 a circuit apparatus, as required in the present invention. Claim 1, by comparison, requires that the threads are hardware apparatus. Specifically, claim 1 states: “each processor having multiple thread units...”, which necessarily implies that the thread units comprise hardware apparatus. This aspect of the invention is described at page 6, lines 4-8, which state that “Each thread unit 204 incorporates a register file (Rfile) 206, a program
20 counter, an arithmetic logic unit (ALU) 207, and logic for instruction fetching....” These components are necessarily hardware apparatus. In the present invention for the thread units to be hardware threads; they cannot be software or execution threads.

The execution threads of Gamo cannot be implemented as hardware. The execution threads must be adaptable to handle a wide variety of commands. For this
25 reason, the execution thread cannot be implemented as hardware, as required in the present invention. The hardware that would be required to accommodate the varied functionality of the execution threads in Gamo would be impractically large and complex. For this reason Gamo cannot be modified to implement the execution thread in hardware. Accordingly, the rejection of claim 1 must be withdrawn for this additional
30 reason.

All claims pending in the application depend from claim 1 and should be allowable for the same reasons noted above.

Regarding claim 20, the Office Action argues that the combination of Irwin and Hartmann would render it obvious to combine four processors as specifically set forth in claim 20 (with two processors providing outbound processing and two processors providing inbound processing). With this arrangement, each processor performs a single type of processing (inbound or outbound for a single protocol) that provides incomplete conversion of the data frame. Consequently, exactly four processors are required, and the apparatus converts between exactly two protocols. It is important to note that, in claim 20, each processor provides only inbound or outbound processing, not both. Col.2, lines 37-42 of Hartmann, by comparison, teach that processors are each fully capable of converting to and from a generic type of protocol (see col. 2, line 41). Any number of processors can be provided, with each processor converting to and from the generic protocol. The processors of Hartmann cannot be modified to operate as set forth in claim 20 because the processors of Hartmann must each convert to and from the generic protocol. This is very different from the present invention as set forth in claim 20 and so the rejection of claim 20 must be withdrawn.

Regarding paragraph 55 of the Office Action, Irwin in col. 2, lines 7-8 and 38-42 does not teach or suggest protocol conversion. In these sections Irwin describes (1) the operation of a forwarding program that controls data traffic, and (2) how data processing is divided among several different processors. Irwin does not teach protocol conversion in a multiprocessor apparatus.

Regarding paragraph 58 of the Office Action, Irwin does not teach storing protocol information. Irwin teaches that header information is used for packet forwarding, but nowhere does Irwin suggest that the type of protocol can be determined by the processor.

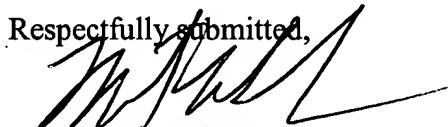
In view of the foregoing, it is respectfully requested that the application be reconsidered, that claims 1-17 and 20 be allowed, and that the application be passed to issue.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone

number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

A provisional petition is hereby made for any extension of time necessary for the continued pendency during the life of this application. Please charge any fees for such provisional petition and any deficiencies in fees and credit any overpayment of fees for the petition or for entry of this amendment to International Business Machines Deposit Account No. 09-0456.

Respectfully submitted,



Michael E. Whitham
Reg. No. 32,635

Whitham, Curtis, & Christofferson, P.C.
11491 Sunset Hills Road, Suite 340
Reston, VA, 20190
Phone: 703-787-9400
Fax: 703-787-7557